

CLAIM REVISIONS

- 1 1. (previously presented) A method of determining the flow of a data object in a software
2 architecture using queues to organize the transfer of data from one processing object to another,
3 comprising :
4 storing a queue indicator in a path object corresponding to a respective data object;
5 receiving and processing the data object in a first of said processing objects;
6 identifying a queue corresponding to a second of said processing objects the identifying
7 depending on the indicator in the path object corresponding to said data object;
8 placing said data object in the queue identified in said step of identifying.

2. (previously presented) A method as in claim 1, wherein said step of identifying includes
determining a result of said step of processing.

3. (previously presented) A method as in claim 1, wherein said step of identifying includes
determining a result of said step of processing; and said queue corresponding to said result.

4-5. (cancelled)

- 1 6. (previously presented) A pipeline software architecture in which data objects are transferred
2 from a first processing object to a selected one of second and third processing objects by queuing
3 the data objects in a queue of said selected one, comprising:
4 a path object corresponding to each of said data objects;

CLAIM REVISIONS

5 at least one of said path objects containing an indicator of at least one of said second and
6 third processing object;

7 said first processing object defining a process a result of which is to insure that a first
8 data object processed by said first processing object is placed in a queue of said at least one of
9 said second and third processing objects responsively to one of said path objects corresponding
10 to said first data object.

1 7. (previously presented) An architecture as in claim 6, wherein said process includes the
2 generation of an indication of a result of processing of said first processing object and said first
3 data object processed by said first processing object is placed in said queue of said at least one of
4 said second and third processing objects responsively to the processing object indicator in the at
5 least one of said path objects corresponding to said first data object and responsively to said
6 result indication.

8-9. (canceled)

10. (previously presented) The method of claim 1, wherein the path object includes a table of
queue indicators.

1 11. (previously presented) The method of claim 1, wherein
2 the processing comprises determining a normal or faulty outcome state of the data object;
3 and
4 the identifying is dependent on said normal or faulty outcome state.

CLAIM REVISIONS

1 12. (currently amended) A method comprising:

2 defining objects, each comprising both data and functions that access the data, the objects

3 including: data objects, and path objects and processing objects;

4 first queuing a data object in a queue of a first processing object in response to a-an

5 indication of the first processing object in a path object associated with the data object;

6 responsive to the first queuing, processing the data object with the first processing object;

7 second queuing the data object in a queue of a second processing object in response to

8 both: results of the processing; and an indication of the second processing object in the path

9 object associated with the data object;

10 responsive to the second queuing, processing the data object with a second processing

11 object.

1 13. (currently amended) Apparatus comprising:

2 objects, each object comprising both data and functions that access the data, the objects

3 including: data objects and path objects and processing objects, each path object mutually

4 corresponding to a respective data object;

5 a respective processing queue for each processing object, the processing objects each

6 process each data object previously queued in the respective queue, the processing of the data

7 object including using the functions of the data object to access the data of the data object, the

8 path objects each comprising indicators of next processing objects for subsequent processing of

9 the corresponding data object after the processing of the data object by the current processing

10 object, the current processing object communicating with the path object to determine the next

CLAIM REVISIONS

11 | processing objects for subsequent processing of the data object, after the processing of the data
12 | object by the current processing unit is complete the processing object queues the data object in a
13 | queue of one of the next processing objects depending on a result of the processing of the data
14 | object by the current processing object.

1 | 14. (new) A medium, readable by a data processing device and embodying data for realizing at
2 | least one realized object for use in an object-oriented programming environment, the
3 | environment comprising processing objects and at least one data object, the at least one data
4 | object needing to be processed along a path via processing objects that are relevant to the data
5 | object, at least three of the processing objects being relevant, the at least one realized object
6 | comprising:

7 | a path object corresponding to the data object, the path object comprising at least three
8 | queue indicators, each queue indicator indicating a respective one of the relevant processing
9 | objects, so that the queue indicators define the path.

1 | 15. (new) The medium of claim 14, wherein the queue indicators comprise a plurality of result
2 | dependent indicators for at least one of the relevant processing objects, the result dependent
3 | indicators including a normal path indicator and a fault path indicator, so that the one relevant
4 | processing object — responsive to its own processing result — can chose a subsequent
5 | processing object for the data object from a group of processing objects designated by the result
6 | dependent indicators in the path object for that data object.

CLAIM REVISIONS

16. (new) The medium of claim 15, wherein the subsequent processing object is indicated by the fault path indicator.

17. (new) The medium of claim 14, wherein the path object comprises an array of queue indicators.

18. (new) The medium of claim 14, wherein the environment comprises a plurality of data objects and at the least one realized object comprises a respective path object for each data object.

CLAIM REVISIONS

Claim 12 has been amended to change "a" to --an--. Similarly the word "of" has been added to claim 13. Applicant respectfully submits that these amendments do not change the scope of the claim, are purely stylistic in nature rather than substantive, and are not in response to any rejection. Applicant accordingly respectfully submits that no filewrapper estoppel should result.

Section 112 rejection: claim 4

This claim has been cancelled, along with its dependent claim. Applicant reserves the right to reassert the claim at a later date if it appears advantageous in the subsequent course of prosecution.

The Examiner seems to think that claims 12 and 13 recite first and second data objects, like claim 4. Applicant is not seeing this recitation. Applicant accordingly respectfully submits that the Examiner has improperly grouped claims 12 and 13 with claim 4 in this rejection. Withdrawal of the rejection against these claims is accordingly respectfully requested.

Art rejections

The art rejections are respectfully traversed.

Since the references are many and/or complex, Applicant will confine his remarks to those portions of the references cited by the Examiner, except as otherwise indicated. Applicant makes no representation as to the contents of other portions of the references.

Any of the Examiner's rejections and/or points of argument that are not addressed below would appear to be moot in view of the following. Nevertheless, Applicant reserves the right to